1984 7/28(二)

PC-8801mkII & PC-8801

dB-I BASIC

REFERENCE HANDBOOK

dB-SOFT

はじめに

dB-IBASICは、dB-SOFTの技術陣が独自の言語思想の統一をめざして開発したPC-8801、およびPC-8801 mk II用のインテジャー（整数型）BASICです。

この言語は、ゲームやホビーのプログラムを作る上で特に威力を発揮することを第1の目的として、組み立てられています。約38Kバイトのユーザーズエリアを持ち、疑似PCGおよびスピード処理が、プログラミングにおける大きな特長といえます。

また、他ハード用として既に発売している実数型のdB-BASICと70%以上のコンパチビリティーを持ち、このdB-BASICの豊富なソフトウェア遺産を継承することができます。

さらに、dB-コンパイラによってdB-IBASICのプログラムを簡単にマシン語に変換することができます。

このdB-IBASICを使いこなして、オリジナル高速ゲームやホビープログラムなどに、ぜひご挑戦ください。ソフトウェアに必ずや新しい可能性を発見するはずです。

最後に、PC-8801という魅力的なハードの生みの親である日本電気株式会社に、心よりお礼申しあげます。

1984年2月1日

古谷貞行

dB-IBASICの起動方法

① 本体の電源を入れます．
② How many files？ で１を入力します．
　　How many files？１⏎
　ここでN88-BASICのタイトルメッセージが表示されます．
③ CLEAR, &H97FF⏎と入力し，dB-IBASICのエリアを確保します．
④ MON⏎と入力し，モニタモードにします．
⑤ テープレコーダにdB-IBASICのテープをセットします．
⑥]R2：⏎と入力し，テープからdB-IBASICをロードします．
⑦ ロード終了後，]G9800⏎と入力します．dB-IBASICが起動します．

dB-IBASICの定数の取扱い

定数は大別して，文字定数と数値定数があります．

1．文字定数

◦ 引用符（〃）で囲まれた255文字以下の英数字，カナ文字，記号，グラフィックキャラクタなどの文字列（ストリング）．
◦ 特に無の文字列をヌルストリング（空文字列）と呼びます．（キャラクタコードで０）
　EX. 〝アイウ〞，〝１２３〞，〝dB-SOFT〞

2．数値定数

数値定数は，整数型です．10進形式と16進形式の表示方法があります．
①10進形式
◦ −32768 ～ ＋32767のすべての整数
　EX.　PRINT　SQR(３)
　　１
　　（$\sqrt{3}$＝1.7320508…）
②16進形式
◦ 16進数を０～９，Ａ，Ｂ，Ｃ，Ｄ，Ｅ，Ｆ で表記します．
◦ 頭に&Hを伴った，最大４桁までの数値です．

dB-IBASICの各演算子の取扱い

1．算術演算

①	∧……指数演算	a ∧ x	(5^3)
②	−……負号	− a	(−32)
③	＊，／(または¥)…乗算，除算	a ＊ b，a／(¥) b	(５×３，20÷５)
④	＋，−……加算，減算	a ＋ b，a − b	(２＋３，６−１)

以上の順で，実行の優先順位となります．

2．整数部の除算と剰余

１）整数としての除算は／（または¥）で行い，商の整数部のみ有効となります．
　EX.　PRINT　10／3
　　３
　　（10÷３＝3.3333……）
２）剰余はMODを用います．
　EX.　PRINT　10 MOD 3
　　１
　　（10÷３＝３余り１）

3. 論理演算

2進の各ビットごとに，0または1を結果として与える．

① AND （＝＊）（論理積，AN.）

X	Y		X AND Y （X＊Y）
1	1	⟶	1
1	0	⟶	0
0	1	⟶	0
0	0	⟶	0

```
EX.   PRINT HEX$(&HFF00 AND &HAAAA)
      AA00
```

```
      1111111100000000  B
AND)  1010101010101010  B
        ↓
      1010101000000000  B
      A  A  0  0
```

② OR （＝＋）（論理和）

X	Y		X OR Y （X＋Y）
1	1	⟶	1
1	0	⟶	1
0	1	⟶	1
0	0	⟶	0

```
EX.   PRINT HEX$(&HFF00 OR &HAAAA)
      FFAA
```

```
      1111111100000000  B
OR)   1010101010101010  B
        ↓
      1111111110101010  B
      F  F  A  A
```

③ XOR （排他的論理和，exclusive or，X.）

X	Y		X XOR Y
1	1	⟶	0
1	0	⟶	1
0	1	⟶	1
0	0	⟶	0

```
EX.   PRINT HEX$(&HFF00 XOR &HAAAA)
      55AA
```

```
      1111111100000000  B
XOR)  1010101010101010  B
        ↓
      0101010110101010  B
      5  5  A  A
```

dB-IBASICの変数の取扱い

変数

1）数値変数

数値変数に型はなく整数型だけですので，DEFINT，DEFDBL，DEFSNG 等の命令文はありません．代入できる整数値は－32768 ～ ＋32767 です．小数部は切り捨てとなります．

2）文字変数

文字変数は変数名の後に〝$〟をつけたもので格納できる文字列（ストリング）は、デフォルト（初期設定）値が15文字までとなっています．この文字数を変更する時は，コマンド$Lで行います．$L⏎を実行すると，変更する文字数をたずねてきますので，最大255文字以内で入力してください．

その他

ファンクションキーの初期定義内容

f・1	save
f・2	auto
f・3	goto
f・4	list CR
f・5	run CR
f・6	load
f・7	width
f・8	print
f・9	console
f・10	color

モニタ仕様

N 88-BASICに準拠します．

グラフィック

グラフィックは絶対座標です．

X：0～639

Y：0～199

コマンド

APPEND

【省略形】 A.
【機　能】 メモリ内プログラムにカセットテープファイル内プログラムを追加する.
【書　式】 APPEND〔"ファイル名"〕
【文　例】 APPEND "TEST" ⏎
【解　説】 ◦メモリ内プログラムの最終行番号より，カセットテープのプログラムの先頭番号が，必ず大きくなければなりません.
◦メモリのテキストの空エリアがアペンドするプログラムより，十分大きくなければなりません.

AUTO

【省略形】 AU.
【機　能】 行番号の自動発生（#はREM発生モードのフリップフロップスイッチ).
【書　式】 AUTO〔#〕〔開始行番号〕〔，増分〕
【文　例】 AUTO #100 ， −10⏎
【解　説】 ◦AUTO実行中もスクリーンエディットが可能.
◦増分に負の値が可能.

BOOT

【省略形】 BO.
【機　能】 dB-IBASICを離れ，N88-BASICのインタプリタレベルに戻る.
【書　式】 BOOT
【文　例】 BOOT⏎
【解　説】 ◦dB-IBASICのインタプリタレベルからN88-BASICのインタプリタレベルに戻り，この時再びdB-IBASICのインタプリタレベルに実行を移したい場合は，特殊コマンドの「CMD」を入力する.
☞ただし，CMDはN88-BASICのインタプリタレベルでのみ有効.
BOOTによるN88-BASICの実行は保障できません。正常に作動しない場合がありますので，N88-BASICだけ使用する時はリセットスイッチ等でN88-BASICを最初から起動してください.

CONT

【省略形】 C.
【機　能】 中断したプログラムの実行再開.
【書　式】 CONT
【文　例】 CONT⏎

DELETE

【省略形】 DE.
【機　能】 プログラムの部分的削除.
【書　式】 DELETE〔第1行番号〕〔{，／−}〔第2行番号〕〕
【文　例】 DELETE 200−300 ⏎
【解　説】 ◦指定された範囲を削除します.
◦第1行番号と第2行番号の指定は，どちらが大きくても構いません.

EDIT

【省略形】 ED.
【機　能】 指定行を画面表示し，プログラム文を編集する.
【書　式】 EDIT {行番号／.}
【文　例】 EDIT 360⏎

FIND

【省略形】 FI.

【機　能】 指定された文字列をプログラムの全ての行，または指定した行番号の中から捜し出し，ディスプレイまたはプリンタに出力する．

【書　式】 FIND〔#〕"文字列"，〔第1行番号〕{ , / − }〔第2行番号〕〕
#：プリンタ出力．

【文　例】 FIND "GOSUB 2000" ⏎

KEY LIST

【省略形】 K. L.

【機　能】 ファンクションキーの定義内容を表示する．

【書　式】 KEY LIST

【文　例】 KEY LIST⏎

LIST／LLIST

【省略形】 L.／LL.

【機　能】 メモリ内のプログラムの全部，または一部を表示・出力する．

【書　式】 { LIST / LLIST } 〔 { 〔行番号1〕 / · } 〕〔 { , / − } 行番号2〕 ……画面 ……プリンタ

【文　例】 LIST 100 − ⏎　　（スペースキーで一時停止）

【解　説】 行番号1と行番号2の指定はどちらが大きくても構いません．

LOAD

【省略形】 LO.

【機　能】 dB-IBASICのプログラムを，カセットテープからメモリに読み込む．

【書　式】 LOAD〔"ファイル名"〕

【文　例】 LOAD ⏎

【解　説】
- dB-IBASICにはファイルディスクリプタがありません．（CAS：等）
- ファイル名は13文字までです．
- 右上隅で "＊" は点滅しません．
- 初期値は600ボーです．

☞600/1200の切換え方法
POKE &H9806,　0 …… 600ボー
POKE &H9806,　1 ……1200ボー
なお、レコーダのコンディションによっては，正常にロードできない場合がありますのでご注意ください．（特に1200ボー）

LOAD ?

【省略形】 LO. ?

【機　能】 カセットテープファイルのチェックサムを照合し，確認する．（ヴェリファイ）

【書　式】 LOAD ?〔"ファイル名"〕

【文　例】 LOAD ?⏎

【解　説】 右上隅で "＊" は点滅しません．

MON

【省略形】 M.

【機　能】 機械語モニタへ制御を移す．

【書　式】 MON

【文　例】 MON ⏎

NEW

【機　能】 メモリにあるプログラムを抹消し，全ての変数をクリアする．

【書　式】 NEW

【文　例】 NEW ⏎

RENUM

【省略形】 REN.
【機　能】 プログラムの行番号を変更，整理する．
【書　式】 RENUM〔<新行番号>〕〔,<旧行番号>〕〔,<増分>〕
【文　例】 RENUM ⏎

RUN

【省略形】 RU.
【機　能】 メモリ内にあるプログラムの実行を始める．
【書　式】 RUN〔<行番号>〕
【文　例】 RUN ⏎

SAVE

【省略形】 SA.
【機　能】 メモリ内にあるプログラムを，ファイルとしてカセットテープに記録する．
【書　式】 SAVE "ファイル名"
【文　例】 SAVE "DATA"
【解　説】 ◦ファイル名は13文字までで，省略はできません。
◦dB-IBASICにはファイルディスクリプタがありません。（CAS：等）
◦初期値は600ボーです．
☞600／1200の切換え方法
POKE &H9806, 0 …… 600ボー
POKE &H9806, 1 ……1200ボー
なお，レコーダのコンディションによっては，正常にセーブできない場合がありますのでご注意ください．（特に1200ボー）

特殊コマンド

CMD

【機　能】 N88-BASICインタプリタレベルから，dB-IBASIC インタプリタレベルに戻る．
【書　式】 CMD
【文　例】 CMD ⏎
【解　説】 ◦dB-IBASIC上でBOOTを行い，N88-BASICに移りますが，その後，再度dB-IBASICインタプリタレベルに戻ることができます．
☞なお，N88-BASICの時にdB-IBASICのインタプリタやワークを破壊してしまった場合，「CMD」でdB-IBASICに戻っても正常に作動しません．

$L

【機　能】 文字変数の文字数を設定する．
【書　式】 $L
【文　例】 $L ⏎
【解　説】 ◦ダイレクト・コマンドモードで実行します．
◦文字列の長さを最大255まで入力できます．
◦文字数の初期値は15文字です．
◦区切りに1バイトとるので，取扱う文字数は（指定数－1）文字となります。
◦オブジェクト効率は2^n-1の値に設定するとよい．（3，7，15，31，63，127，255）

一般ステートメント

CANCEL

【省略形】 CAN.
【機　能】 サブルーチンにおいてRETURN文により呼出したプログラムへの復帰を取り消す.
【書　式】 CANCEL
【文　例】 CANCEL：GOTO 100

CASE END

【省略形】 CAS. E.
【機　能】 CASE文に続く一連の条件判定処理を終了する.
【書　式】 CASE END
【文　例】 CASE END
【解　説】 ◦判定処理が全て終了したらCASE ENDを入れ、CASE条件判定モードを解除します.
◦CASE判定から外へ分岐（GOTO, GOSUB）する時は，必ずCASE ENDを行います.

CASE TRUE OF

【省略形】 CAS. TR. OF
【機　能】 条件判断を行い，真の時，以下に続く実行を行う.
【書　式】 CASE TRUE OF
【文　例】 CASE TRUE OF
A＝1：B＝0
⋮　　⋮
CASE　END
【解　説】 ◦左側の条件が正しい（真）場合，次の命令（右側）を実行し，違う場合はその次の行に進みます.
◦ELSEを条件として使うことはできません.

CASE FALSE OF

【省略形】 CAS. FA. OF
【機　能】 条件判断を行い，偽の時，以下に続く実行を行う.
【書　式】 CASE FALSE OF
【文　例】 CASE FALSE OF
A$＝〝Y〞：PRINT〝マチガイ〞
⋮　　⋮
CASE END
【解　説】 ◦左側の条件が違う（偽）場合，次の命令（右側）を実行し，真の場合はその次の行に進みます.
◦ELSEを条件として使うことはできません.

CLEAR

【省略形】 CLE.
【機　能】 すべての変数の初期化とユーザーズ領域の設定をする.
【書　式】 CLEAR BASICテキスト上限アドレス
【文　例】 CLEAR &H77FF

CLR

【機　能】 すべての変数の初期化を行う。
【書　式】 CLR
【文　例】 CLR

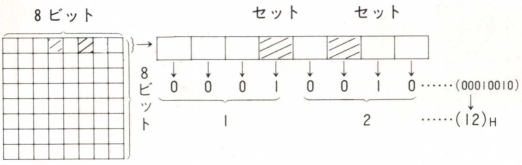
DATA

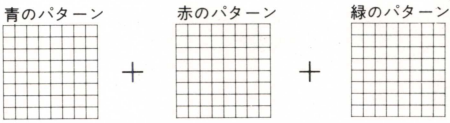
【省略形】 D.
【機　能】 READで読み込まれる数値，および文字定数を定義する．
【書　式】 DATA 定数〔，定数…〕
【文　例】 DATA 1，10，dB，PC

DEFCHR$

【省略形】 DEF.
【機　能】 疑似PCGにキャラクタ・パターンデータを定義する．
【書　式】 DEFCHR$(キャラクタコード) ＝〈パターンデータストリング〉
【文　例】 DEFCHR$(32)＝HEXCHR$(STRING$(24,"00"))
【解　説】
- 青，赤，緑，それぞれの色ごとにパターンを指定します．パターンは横方向8ドットを1バイトデータで指定し，1つのパターンデータは8バイトデータで構成され，3色（青，赤，緑）指定するので1つのキャラクタに対するパターンデータは24バイトデータとなります．
- キャラクタパターンは縦横8×8ドットです．横8ドットが1バイトデータです．この時，パターンセットするところが1，それ以外が0とし，2進数で考えると8ビットデータとなり16進数で表すと2桁の1バイトデータとなります．　この16進のデータがそれぞれ横8ドットのデータに対応しますので，1つのキャラクタパターンは8バイトの16進データからなります．
- PCGでは青，赤，緑，それぞれがキャラクタパターンを持っているので，この8バイトの16進データが各色分必要です．8バイトデータ＋8バイトデータ＋8バイトデータ，計24バイトのデータからPCGによる1キャラクタのパターンデータは構成されます．3色の合成ですので，結果的にキャラクタパターンは1ドットに対し8色から1色選択できます．

☞ 疑似PCGを使う場合は，必ずPCG定義前にCLEAR &H77FFを実行し，PCGデータエリアを確保してください．
PCGデータエリア内にテキストウインドウがありますので，PCGデーターが失われる場合があります．





DIM

【省略形】 DI.
【機　能】 配列変数の要素の大きさを宣言し，メモリに割りあてる．
【書　式】 DIM〈変数名（添字の最大値)〉〔,〈変数名(添字の最大値)〉…〕
【文　例】 DIM A(40，10)，B$(200)
【解　説】 2次元までの配列で，添字は0～254までです．

END

【省略形】 E.
【機　能】 プログラムの終了を宣言する．
【書　式】 END
【文　例】 END

FOR…TO…STEP～NEXT

【省略形】 F.…TO…STE.～N.

【機　能】 プログラムの指定された部分を繰り返し実行する.

【書　式】 FOR 制御変数＝式1(初期値) TO 式2(最終値)〔STEP 式3(増分)〕
NEXT〔制御変数〕

【文　例】
```
FOR I＝0 TO 3
FOR J＝0 TO 3
READ DT$ (I, J)
NEXT J
NEXT I
```

【解　説】
- dB-IBASICでは，N88-BASICと違い，式1の値が式2の値になる前に，このループから抜け出る事はできません.
- 増分に数値変数は使えません.

GOSUB

【省略形】 GOS.

【機　能】 BASICプログラムのサブルーチンを呼び出す.

【書　式】 GOSUB 行番号またはラベル

【文　例】 GOSUB "SUBROUTINE"

GOTO

【省略形】 G.

【機　能】 指定したプログラム行番号へプログラムの実行を移す.

【書　式】 GOTO 行番号またはラベル

【文　例】 GOTO 20

IF～THEN…(ELSE)／IF～GOTO/GOSUB…(ELSE)

【省略形】 IF～TH.…EL.／IF～G./GOS.…EL.

【機　能】 論理式の条件判断を行う.（文の時，THENは省略可）

【書　式】
```
IF 論理式 | 〔THEN〕{文 / 行番号 / "ラベル名"} | 〔ELSE {文 / 行番号 / "ラベル名"}〕
          | {GOTO / GOSUB} {行番号 / "ラベル名"} |
```

【文　例】 IF A$＝CHR$(13) THEN CLS ELSE "INPUT"

ON～GOSUB

【省略形】 O. GOS.

【機　能】 式の値により，サブルーチンの選択を行う.

【書　式】 ON 式 GOSUB サブルーチンの飛び先（行番号またはラベル）〔,サブルーチンの飛び先……〕

【文　例】 ON SN GOSUB 100, 200, 300, 400, 500

ON～GOTO

【省略形】 O. G.

【機　能】 飛び先の選択を行う.

【書　式】 ON 式 GOTO 飛び先（行番号またはラベル）〔,飛び先…〕

【文　例】 ON N GOTO 100, 200, 300

READ

【省略形】 REA.

【機　能】 DATA文で定義された値を変数に読み込む.

【書　式】 READ 変数名〔,変数名…〕

【文　例】 READ A, B$

REM

【省略形】 7 ▼ ヤ キーの ▼ で代用できる.

【機　能】 プログラムに注釈を入れる非実行文.

【書　式】 REM〔〈注釈文字列〉〕

【文　例】 REM＊＊コメント

▼　ニュウリョク

RESTORE

【省略形】 RES.

【機　能】 READ 文で読む DATA 文を指定する.

【書　式】 RESTORE〔{行番号 / "ラベル名"}〕

【文　例】 RESTORE "ゲームセツメイデータ"

RETURN

【省略形】 R.

【機　能】 サブルーチンを終了し，GOSUB が行われた次のプログラム文，または次の行番号へプログラムの実行を戻す.

【書　式】 RETURN

【文　例】 RETURN

PEPEAT～UNTIL

【省略形】 REP.～ U.

【機　能】 REPEAT と UNTIL の間を条件が満足するまで繰り返し実行する.

【書　式】 REPEAT ～ UNTIL　条件式

【文　例】 REPEAT

K$＝INKEY$（0）

UNTIL　K$<>""

【解　説】 ◦REPEAT～UNTIL 間では，他の処理への分岐はできません.

STOP

【省略形】 S.

【機　能】 プログラムの実行を中断してコマンドレベル（コマンド入力待ちの状態）に戻る.

【書　式】 STOP

【文　例】 STOP

＊"ラベル名"

【機　能】 GOTO，GOSUB 等の参照先をラベル名で定義する.

【書　式】 行番号 ＊"〈ラベル〉"

【文　例】 100 ＊"DATA テーブル"

【解　説】 ◦文字列を使い，参照先を設定します.

◦行番号の次に指定し，必ずアスタリスク（＊）を付け，ダブルコーテーション（"）でラベル名（文字列）をくくります.

◦ラベルの長さは，プログラムの 1 行の許容範囲（1 行255文字）によってのみ制限を受けます.

入出力ステートメント

INPUT

【省略形】 I.
【機　能】 キーボードより，指定した変数へ入力する．
【書　式】 INPUT〔〝プロンプト文〞；〕変数
【文　例】 INPUT〝ナマエ〞；A$

MEM

【省略形】 ME.
【機　能】 指定されるアドレスから，定数の値をメモリに書き込む．
【書　式】 MEM { 式 / # }，定数〔，定数…〕
【文　例】 MEM &H6000，&H21，&H22，&H23
【解　説】 ◦式……定数データを書き込み始める先頭アドレスを指定．
◦#……直前のMEM文によって書き込まれたデータの最終アドレス＋1のアドレスを，書き込み開始アドレスとして指定．

OUT

【省略形】 OU.
【機　能】 指定された出力ポートに8ビット（1バイト）のデータを送る．
【書　式】 OUT ポート番号，1バイトデータ
【文　例】 OUT 81,33：BEEP 1：OUT 81,32

POKE

【省略形】 PO.
【機　能】 メモリ上の指定番地へ数値を書き込む．
【書　式】 POKE アドレス，数値
【文　例】 POKE &HF3C8，&H41

POKE @

【省略形】 PO. @
【機　能】 G-VRAMに1バイトのデータを送る．
【書　式】 POKE @ アドレス，出力データ
【文　例】 POKE @ &H8000，&HFF
【解　説】 G-VRAMアドレス（仮想的）
青：0～3FFF(H)
赤：4000～7FFF(H)
緑：8000～BFFF(H)

POKW

【機　能】 POKEの2バイト命令．
【書　式】 POKW〔@〕書き込み番地，2バイトデータ
【文　例】 POKW &H7000，&H3020

PRINT

【省略形】 P. または ?

【機　能】 ディスプレイに情報を出力する.

【書　式】 PRINT〔<式>{ ' / ; }〔<式>〕…〕〕

【文　例】 PRINT "SCORE" ; 123, A * 10

PRINT #0

【省略形】 P. #0 または ?#0

【機　能】 コントロールコードの非実行のキャラクタ，および式を表示する.

【書　式】 PRINT#0, 式〔{ ' / ; } 式…〕

【文　例】 FOR J= 0 TO 31 : PRINT#0, CHR$(J) ; : NEXT J

スクリーンステートメント

BOX

【機　能】 指定したテキスト画面上に，指定したキャラクタの箱を描く.
(位置は箱の対角線をテキスト座標で指定する)

【書　式】 BOX (水平方向位置，垂直方向位置) − (水平方向位置，垂直方向位置)，1 文字

【文　例】 BOX (8, 10) − (25, 20), "A"

CIRCLE

【省略形】 CI.

【機　能】 画面上の任意の座標を中心点として楕円 (円)，または円弧，または多角形を描く.

【書　式】 CIRCLE (X, Y)，横の半径，〔色〕，〔縦の半径〕，〔開始角度〕，〔終了角度〕，〔ステップ角度〕

【文　例】 CIRCLE (300, 100), 80, 1, 50, 0, 3000, 87

CLS

【省略形】 CL.

【機　能】 画面表示を消去し，カーソルをホームポジションに戻す.

【書　式】 CLS〔式〕　式 { 0 ：全グラフィック / 1 ：ページ1 / 2 ：ページ2 / 3 ：ページ3 / 4 ：全グラフィックおよびテキスト / 省略：テキストのみ }

【文　例】 CLS 3

CONSOLE

【省略形】 CONS.

【機　能】 テキスト画面の，表示モードの設定を行う.

【書　式】 CONSOLE〔スクロール開始，スクロール行数〔，ファンクションキー表示スイッチ〕〔，カラー／白黒スイッチ〕〕

【文　例】 CONSOLE 0, 10, 0, 1

【解　説】 ◦ファンクションキー表示スイッチ { 0：OFF / 1：ON }

◦カラー／白黒スイッチ { 0：白黒モード / 1：カラーモード }

COLOR

【省略形】 COL.

【機　能】 ディスプレイの各部の色を指定する.

【書　式】 COLOR〔ファンクションコード〕〔, グラフィックカラー〕〔, ボーダーカラー〕

【文　例】 COLOR 7, 7, 0

【解　説】 ◦ファンクションコード（テキスト画面に対して行う）

白黒モードの場合（CONSOLE,,, 0）

0	1	2	3	4	5	6	7
ノーマル	シークレット	ブリンク	シークレット	リバース	リバース シークレット	リバース ブリンク	リバース シークレット

カラーモードの場合（CONSOLE,,, 1）

0	1	2	3	4	5	6	7
黒	青	赤	紫	緑	水色	黄	白

◦グラフィックのカラー，およびボーダーカラーも0～7までのカラーコードで指定します.

LOCATE

【省略形】 LOC.

【機　能】 画面上のカーソルの位置を指定する.

【書　式】 LOCATE 水平方向位置, 垂直方向位置

【文　例】 LOCATE 10, 10：PRINT "ABC"

LINE

【省略形】 LIN.

【機　能】 グラフィックスクリーンの指定された2点間に直線，または箱を表示する.

【書　式】 LINE〔(始点座標)〕－(終点座標)〔, { PSET / PRESET / XOR }〔, { カラーコード / ラインパターン }〕〔{ B / BF }〕〕〕

【文　例】 LINE(100, 100)－(500, 150), PSET, "ABC", BF

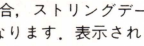
【解　説】 ◦座標はそれぞれ（X座標，Y座標）の型で指定します.

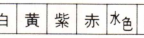
◦PSETは指定色で線を描き，PRESETは線を消します．また，XORは線の部分を反転します.

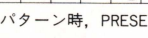
◦カラーコード省略時は，直前のCOLOR文の指定色（グラフィックカラー）となります.

◦ラインパターンはストリングデータで，横方向に8ドットごと指定します．青，赤，緑の順で各1バイトとなります.

EX.（▨のドットがセット）

青  10101010 (2) ＝ AA (H)

赤  11110000 (2) ＝ F0 (H)

緑  11001100 (2) ＝ CC (H)

この場合，ストリングデータはHEXCHR$（"AAF0CC"）または"ェ×7"となります．表示されるラインパターンは上記3色の重ね合わせで

白	黄	紫	赤	水色	緑	青	黒

となります.

◦ラインパターン時，PRESETはPSETと同じです.

GPRINT

【省略形】 GPR.

【機　能】 疑似PCGで定義したキャラクタを表示する.

【書　式】 GPRINT〔#引数, 〕式〔, 式……〕

【文　例】 DEFCHR$(0)=HEXCHR$("001824C3E75A24420000183C
18240000000018241800000")：GPRINT CHR$(0)

【解　説】
- 引数　0：PSET, 1：XOR, 2：OR
- 式はPRINT文と同様に行います.
- 疑似PCGはグラフィックです.
- 最下行ではPRINT文と違いスクロールしません.

GPUT

【省略形】 GP.

【機　能】 式で示されるアドレスの内容を見て, 座標X, Yで示される画面上にキャラクタコードデータに該当する疑似PCGのキャラクタを表示する.

【書　式】 GPUT (X, Y) {,/−} 式

【文　例】 MEM &H7000, 2, 2, 0, 1, &H10, &H11：GPUT(60, 21)
−&H7000

【解　説】
- X座標……0～79 (0～40のモードなし)
- Y座標……0～199 (ドット単位)
- WIDTH文でPCGキャラクタの大きさは変わりますが、座標は変わりません.
- 式……アドレス
- 式のアドレスの内容……X方向のキャラクタ数
- 式＋1のアドレスの内容……Y方向のキャラクタ数
- 式＋2のアドレス以降……キャラクタのデータ
- 疑似PCGはグラフィックです.

MPUT

【省略形】 MP.

【機　能】 式で示されるアドレスの内容を見て, 座標X, Yで示される画面上にデータに該当するキャラクタをPUTする.

【書　式】 MPUT (X, Y) {,/−} 式

【文　例】 MEM &H7000, 3, 1, &H41, &H42, &H43：MPUT(12, 12)−&H7000

【解　説】
- (X, Y)……テキスト画面の座標
- 式……アドレス
- 式のアドレスの内容……X方向のキャラクタ数
- 式＋1のアドレスの内容……Y方向のキャラクタ数
- 式＋2のアドレス以降……キャラクタのデータ

PALET

【省略形】 PAL.

【機　能】 グラフィックのパレット番号に対して色を割りあてる.

【書　式】 PALET〔パレット番号, カラーコード〕

【文　例】 PALET 7, 1

【解　説】
- パレット番号：0～7
- カラーコード：0～7
- 数値省略時はデフォルト値に設定.

PAINT

【省略形】 PA.

【機　能】 指定された色，またはタイルパターンで指定された場所を塗る．

【書　式】 PAINT(X,Y)〔,{カラーコード / タイルパターン}，境界色〕

【文　例】 PAINT (200, 100), "ABC", &H80

【解　説】
- 座標　X：0～639，Y：0～199
- カラーコード　0～7
- タイルパターンはストリングデータで横方向に8ドットごとに青，赤，緑の順で各1バイトデータで指定します．
- 縦方向は最大8ドットまで（24バイトデータ）です．

EX.　次のタイルパターンで塗る場合

青	青	白	緑	緑	白	赤	赤
紫	紫	白	黄	黄	白	水	水

青11100100(2), 赤00100111(2), 緑00111100(2)→E4273C(H)
青11100111(2), 赤11111100(2), 緑00111111(2)→E7FC3F(H)

よってタイルパターンはHEXCHR$("E4273CE7FC3F")となります．
(PAINT(X, Y), HEXCHR$("E4273CE7FC3F"), &H8C)

- 境界色は1バイトの数値データで，各色に各ビットが対応します．

ビット	7	6	5	4	3	2	1	0
ビットの重み	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
色	白	黄	水色	緑	紫	赤	青	黒

EX.　白と赤と紫が境界色の場合，10001100(2) となり10進で$2^7+2^3+2^2=140$　16進で8C(H) となります．

PATTERN

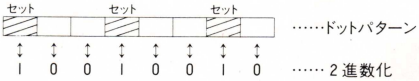
【省略形】 PAT.

【機　能】 グラフィックスクリーンにグラフィックパターンを設定する．

【書　式】 PATTERN 式,(文字列)

【文　例】 PATTERN 2, "AA"

【解　説】
- 式は－255～＋255でY軸（垂直）方向のドット数を指定します．
- 正の場合は開始位置より上方向，負の場合は下方向．
- 文字列でX軸（水平）方向のデータとなる文字を与えます．
- 表示位置はPOSITION文で指定．
- パターンは，X軸方向に8ドット単位で設定され，1文字（8ドット）のデータと対応します．
- 実際のドットのセット
  次のようにドットパターンをセットした場合



- データは，2進数で10010010(2)　16進数で92(H)
- データとなる数字を文字列データとしてPATTERN文で設定します。
- 以上の8ドットのみのセットであればPATTERN 1, CHR$(&H92)

POSITION

【省略形】 POS.

【機　能】 PATTERN文の表示位置を設定する．

【書　式】 POSITION　X座標，Y座標

【文　例】 POSITION 50, 100

PRESET

【省略形】 PRE.
【機　能】 グラフィックスクリーン上の任意のドットをリセットする.
【書　式】 PRESET(X座標, Y座標〔, カラーコード〕)
【文　例】 PRESET(100, 50)
【解　説】 ◦X座標は水平方向のドットの座標：0～639
◦Y座標は垂直方向のドットの座標：0～199

SCREEN

【省略形】 SC.
【機　能】 スクリーンに対し，それぞれの状態を設定する.
【書　式】 SCREEN〔テキスト画面〕〔, 入力可能グラフィック〕
【解　説】 ◦テキスト画面 {0：テキスト画面　ON / 1：テキスト画面　OFF}
◦入力可能グラフィック：0～7（各カラーコード対応）

WIDTH

【省略形】 W.
【機　能】 テキスト画面に表示する文字の桁数，および行数を設定する.
【書　式】 WIDTH〔画面1行の文字桁数〔, 文字行数〕〕
【文　例】 WIDTH 80, 25
【解　説】 ◦文字桁数：40または80
◦文字行数：20または25
◦WIDTHのみの場合は，40/80の切換え.

特殊ステートメント

CALL

【省略形】 CA.
【機　能】 機械語サブルーチンを呼び出す.
【書　式】 CALL 実行番地
【文　例】 CALL &HC000

KEY

【省略形】 K.
【機　能】 キーボード上のプログラマブル・ファンクションキーの内容を定義する.
【書　式】 KEY キー番号, 文字列
【文　例】 KEY 1,"CLS0"＋CHR$(13)
【解　説】 ◦文字列はそれぞれ15文字までです.

USR

【省略形】 US.
【機　能】 直接Z-80（CPU）のレジスタに値を渡して機械語サブルーチンをコールする.
【書　式】 USR 指定アドレス〔, 〔HLレジスタのデータ〕, 〔DEレジスタのデータ〕, 〔BCレジスタのデータ〕, 〔Aレジスタのデータ〕〕
【文　例】 USR &H7000, &H1000, &H2000, &H3000, &H40
【解　説】 ◦ステートメントですから，USRの後に直接，呼び出すアドレスを指定します.

WAIT

【省略形】 WA.
【機　能】 指定した時間だけプログラムの実行を停止する.
【書　式】 WAIT 引数　　(引数 msec ウエイト)
【文　例】 WAIT 2

音制御ステートメント

BEEP

【省略形】 B.
【機　能】 指定された回数だけブザーを鳴らす.
【書　式】 BEEP〔式〕
【文　例】 BEEP 3

SOUND

【省略形】 SO.
【機　能】 指定した長さの音を、指定の回数鳴らす.
【書　式】 SOUND 音の鳴る長さ, 音の止まる長さ, 繰り返す回数
【文　例】 SOUND 1000, 10, 20

一般関数

FRE

【省略形】 FR.
【機　能】 メモリの未使用領域の大きさを与える.
【書　式】 FRE(引数)　　(引数はダミー)
【文　例】 PRINT FRE(0)
【解　説】 テキストの空エリアを与えます. また, 32768 バイト以上は補数表記されます.

PEEK

【機　能】 メモリの指定番地の内容を読み出す.
【書　式】 PEEK(番地)
【文　例】 A＝PEEK(&H3000)

PEEK@

【省略形】 PE.
【機　能】 G-VRAM の値を得る.
G-VRAM のアドレス. (仮想的)
青：0～3FFF(H)
赤：4000～7FFF(H)
緑：8000～BFFF(H)
【書　式】 PEEK@(アドレス)
【文　例】 A＝PEEK@(&H5000)

PEEW

【機　能】　PEEKの2バイト命令.
【書　式】　PEEW[@](番地)
【文　例】　PEEW(&HA000)
【解　説】　◦指定アドレスから，2バイトの内容(データ)を読み取ります.
◦@を付けた場合，G-RAM又はV-RAMのアドレスとなります.
◦Adr(任意の番地)に△□(H)，Adr＋1に○×(H)というデータがそれぞれ入っている場合，PEEW(Adr)を実行すると，○×△□(H)という値で読み込みます.

POINT

【省略形】　POI.
【機　能】　ディスプレイの指定された座標のドットのカラーコードを与える.
【書　式】　POINT(X座標，Y座標)
【文　例】　PRINT POINT(100,100)

POS

【機　能】　テキスト画面上の現在のカーソルの水平方向位置，又は垂直方向位置を与える.
【書　式】　POS(式)　　式 {0：水平方向位置 / 1：垂直方向位置}
【文　例】　PX＝POS(0)

SPC

【省略形】　SP.
【機　能】　現在のカーソルのある行の先頭から任意の位置までスペース(空白)を出力する.
【書　式】　SPC(数式)
【文　例】　PRINT SPC(10);"ABC"

TAB

【省略形】　TA.
【機　能】　現在のカーソルのある行の先頭から任意の位置まであける.
【書　式】　TAB(数式)
【文　例】　PRINT "ABC";TAB(10);"DEF"

数値関数

ABS

【省略形】　AB.
【機　能】　絶対値を与える.　　|X|
【書　式】　ABS(数式)
【文　例】　A＝ABS(－10)

COS

【機　能】　余弦(コサイン)を与える.
【書　式】　COS(式, 倍数)
【文　例】　A＝COS(180,100)
【解　説】　◦式の値は度単位です.
　　◦倍数でCOSの値を何倍にするか指定します.

RND

【省略形】　RN.
【機　能】　0と(式の値)－1の間で乱数を与える.
【書　式】　RND(式)
【文　例】　A＝RND(10)　　→0から9まで.

SGN

【省略形】　SG.
【機　能】　符号を調べる.
【書　式】　SGN(数式)
【文　例】　A＝SGN(B)

SIN

【省略形】　SI.
【機　能】　正弦(サイン)を与える.
【書　式】　SIN(式, 倍数)
【文　例】　PRINT SIN(30,100)
【解　説】　◦式の値は度単位です.
　　◦倍数でSINの値を何倍にするか指定します.

SQR

【省略形】　SQ.
【機　能】　平方根(ルート)を与える.　　$\sqrt{X}$
【書　式】　SQR(数式)
【文　例】　A＝SQR(2)

ストリング関数

ASC

【省略形】　AS.
【機　能】　文字のキャラクタコードを与える.
【書　式】　ASC(文字列)
【文　例】　PRINT ASC("A")

CHR$

【省略形】 CH.
【機　能】 指定したキャラクタコードに対応する文字を与える.
【書　式】 CHR$(キャラクタコード)
【文　例】 PRINT CHR$(&H34)

HEXCHR$

【省略形】 HEXC.
【機　能】 16進数表現の文字列をキャラクタコードとみなし，対応する文字列を与える.
【書　式】 HEXCHR$(16進数の文字列)
【文　例】 A$＝HEXCHR$("64422D534F4654"):PRINT A$
【解　説】 ◦16進数形式表記文字列の先頭から2文字ずつが,1バイトデータとして、対応するキャラクタコードの文字に順次変換し文字列をつくります。
◦スペースは区切りとみなします.

HEX$

【省略形】 H.
【機　能】 10進数を16進数に変換して，その文字列を与える.
【書　式】 HEX$(数式)
【文　例】 A$＝HEX$(128)
【解　説】 ◦返す文字列は，00～FFFF(H)
◦数式は0～65535（又は補数表記－32768～＋32767）

INKEY$

【省略形】 INK.
【機　能】 キーボードから1文字取り込み，その文字を与える.
【書　式】 INKEY$(引数)

引数
- 0：直接キー入力を読み込む.
- 1：ヌルストリングは無視．カーソル点滅.
- 2：キーボードの入力バッファの1バイトデータを読み込む．バッファが空の時は，ヌルストリングを与える.（カーソルなし）

【文　例】 A$＝INKEY$(1)

INSTR

【省略形】 INS.
【機　能】 文字列の中に任意の文字列が含まれているか捜しその位置を与える.
【書　式】 INSTR(文字列1，文字列2)
【文　例】 A＝INSTR(N$，"フルヤ")

LEFT$

【省略形】 LEF.
【機　能】 文字列の左側から任意の長さの文字列を取り出す.
【書　式】 LEFT$(文字列，数式)
【文　例】 A$＝LEFT$(A$,3)

LEN

【省略形】 LE.
【機　能】 文字列の総文字数を算出する.
【書　式】 LEN(文字列)
【文　例】 A＝LEN(A$)

MID$

【省略形】 MID.
【機　能】 文字列の中から任意の長さの文字列を取り出す.
【書　式】 MID$(文字列, 式1, 式2)　　(式1：桁位置　式2：文字数)
【文　例】 A$=MID$(A$, 3, 10)

MIRROR$

【省略形】 MI.
【機　能】 文字列を逆アスキーコードに変換して, その文字列を与える.
【書　式】 MIRROR$("文字列")
【文　例】 PRINT #0, MIRROR$(K$)
【解　説】 指定された文字列の1文字ずつのアスキーコード (キャラクタコード) の2進の8ビットを逆にします. LSBとMSBが逆になります.

RIGHT$

【省略形】 RI.
【機　能】 文字列の右側から任意の長さの文字列を取り出す.
【書　式】 RIGHT$(文字列, 数式)
【文　例】 A$=RIGHT$(AB$, 2)

SCRN$

【省略形】 SCRN.
【機　能】 指定したテキスト画面上に表示されている文字列を与える.
【書　式】 SCRN$(カーソル水平位置, カーソル垂直位置〔, 文字数〕)
【文　例】 A$=SCRN$(0, 5, 3)
【解　説】 文字数省略時は1, 指定値は1～255

SPACE$

【省略形】 SPA.
【機　能】 任意の長さのスペース (空白) を与える.
【書　式】 SPACE$(数式)　(数式はスペースの文字数)
【文　例】 PRINT "ABC";SPACE$(10);"DEF"

STR$

【機　能】 数値を文字列に変える.
【書　式】 STR$(数式)
【文　例】 A$=STR$(A)

STRING$

【省略形】 STRIN.
【機　能】 任意の文字列を任意の数だけ与える.
【書　式】 STRING$(数式, 文字列)
【文　例】 PRINT STRING$(30, "A")

VAL

【省略形】 V.
【機　能】 文字列の表す数値を与える.
【書　式】 VAL(文字列)
【文　例】 A=VAL("123")

タイマー関数(システム変数)

TIME

【省略形】 T.
【機　能】 内蔵クロックの秒数の設定，および秒数のカウントを行う．
【書　式】 TIME
【文　例】 PRINT TIME

特殊関数

STICK

【省略形】 STI.
【機　能】 キーボードのテンキーからの入力値を与える．
【書　式】 STICK(引数)　　(引数はダミー)
【文　例】 S=STICK(1)

STRIG

【省略形】 STR.
【機　能】 キーボードのスペースキーの入力を調べる．(0または−1を返す)
【書　式】 STRIG(引数)　　(引数はダミー)
【文　例】 S=STRIG(2)

エラーメッセージ

エラーメッセージ	意　　味
CANCEL without GOSUB	GOSUB がないのに CANCEL を使った.
Can't continue	CONT コマンドによるプログラムの実行再開ができない.(ポインタの値が壊されている)
CASE Already defined	CASE の中に CASE がある.
CASE not defined	CASE が無い.
Device offline	使用しようとしたI/Oデバイスがつながっていない.(オンラインになっていない)
Duplicate Definition	同じ名前の配列を 2 度宣言している.
Illegal Function call	ステートメントや関数において機能の呼び方が誤っている.
Memory over	メモリ容量が足りなくなった.(プログラムが大きすぎる.スタックを消費尽した.配列が大きすぎる)
Missing operand	ステートメント中に必要なパラメータ(オペランド)が抜けている.
NEXT without FOR	NEXT に対応する FOR 文がない.(入れ子構造が正しくない)
Out of data	READ 文によって読むべきデータがない.
Overflow	演算結果や入力された数値が許される範囲をこえた.
RETURN without GOSUB	GOSUB 文で呼び出されていないのに, RETURN 文が出てきた.
String bufer overflow	文字領域がオーバーフローした.
String Overflow	文字定数, 又は文字変数内の文字数が255文字を超えている.
Subscript out of range	配列変数の添字の値が規定範囲から外れている.
Syntax error	コマンド, 又は文の記述に誤りがある.
Tape read error	テープからの読み込み時にエラーが発生した.
Too complex	文字式が複雑すぎる.(カッコのネスティングが深すぎるなど)
Type Mismatch	式の左辺・右辺, 関数の引数などにおいて変数または定数の形が合わない.
Undefine line	指定された参照先がない.

コントロールコード

キャラクタコード	対応するキー	機　　能
2	CTRL-B	1つ前のワードへ戻る
3	CTRL-C	実行の停止．ストップ・キーと同じ
4	CTRL-D	1ワード削除
5	CTRL-E	1行削除
6	CTRL-F	1つ先のワードへ進む
7	CTRL-G	スピーカを鳴らす
8	CTRL-H	バックスペース
9	CTRL-I	水平タブ．タブキーと同じ
10	CTRL-J	ラインフィード．インサートモードで2行に分割
11	CTRL-K	ホームポジション
12	CTRL-L	テキスト画面クリア
13	CTRL-M	キャリッジリターン
15	CTRL-O	画面への表示を無効にする
18	CTRL-R	インサートモード
19	CTRL-S	表示を一時停止する
21	CTRL-U	1行キャンセル
24	CTRL-X	1行の最後へ進む
28	→	カーソルを右へ移動
29	←	カーソルを左へ移動
30	↑	カーソルを上へ移動
31	↓	カーソルを下へ移動

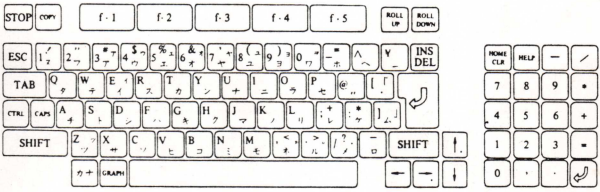
カラーコード

コード	0	1	2	3	4	5	6	7
色	黒	青	赤	マゼンタ	緑	シアン	黄	白

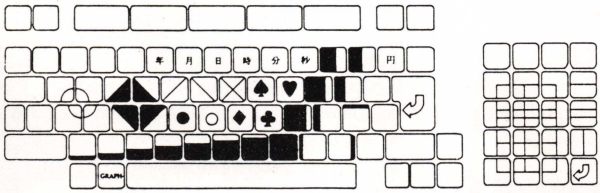
キャラクタコード表

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0		DE		0	@	P		p	▁	┴		―	タ	ミ	╧	╳
1	SH	D1	!	1	A	Q	a	q	▂	┬	。	ア	チ	ム	╨	円
2	SX	D2	"	2	B	R	b	r	▃	┤	「	イ	ツ	メ	╪	年
3	EX	D3	#	3	C	S	c	s	▄	├	」	ウ	テ	モ	╫	月
4	ET	D4	$	4	D	T	d	t	▅	─	、	エ	ト	ヤ	◢	日
5	EQ	NK	%	5	E	U	e	u	▆	━	・	オ	ナ	ユ	◣	時
6	AK	SN	&	6	F	V	f	v	▇	│	ヲ	カ	ニ	ヨ	◥	分
7	BL	EB	'	7	G	W	g	w	▇	┃	ァ	キ	ヌ	ラ	◤	秒
8	BS	CN	(	8	H	X	h	x	▌	┌	ィ	ク	ネ	リ	♠	
9	HT	EM	)	9	I	Y	i	y	▌	┐	ゥ	ケ	ノ	ル	♥	
A	LF	SB	*	:	J	Z	j	z	▌	└	ェ	コ	ハ	レ	◆	
B	HM	EC	+	;	K	[	k	{	▌	┘	ォ	サ	ヒ	ロ	♣	
C	CL	→	,	<	L	¥	l	:	▌	╭	ャ	シ	フ	ワ	●	
D	CR	←	−	=	M	]	m	}	▌	╮	ュ	ス	ヘ	ン	○	
E	SO	↑	.	>	N	^	n	~	▌	╰	ョ	セ	ホ	゛	╱	
F	SI	↓	╱	?	O	_	o		┼	╯	ッ	ソ	マ	゜	╲	

キャラクタキー配列



キーの配列



グラフィックシンボルのキー配列

N88-BASIC／dB-IBASIC対照表

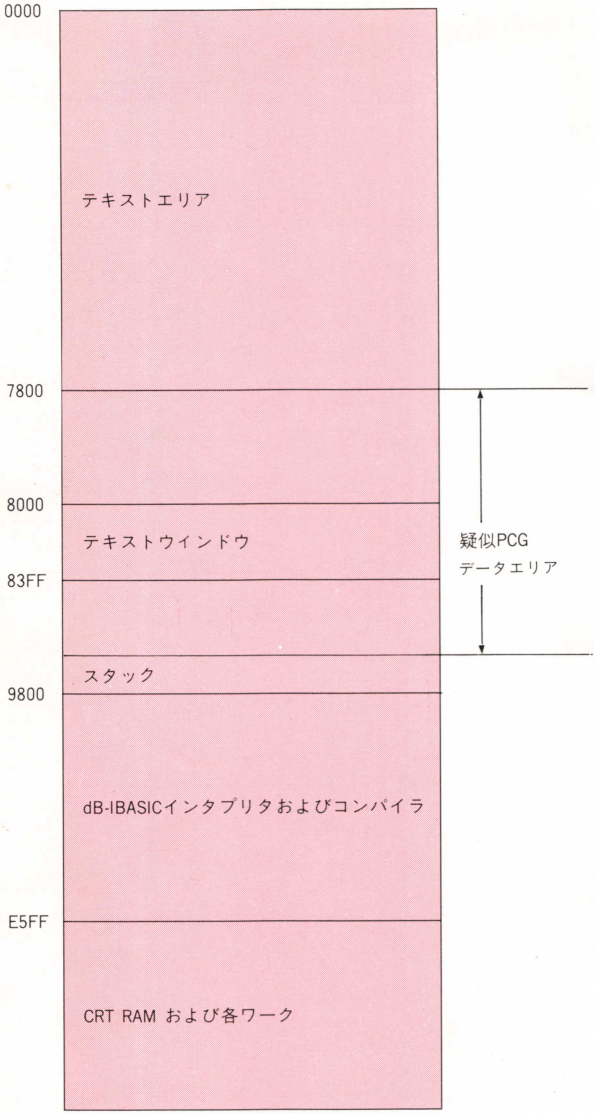
N88-BASIC	dB-IBASIC
AUTO	REM，逆AUTOも可能．
BEEP	BEEP n.　　n回（1回の長さはCTRL-Gと同じ）．
CIRCLE	書式が違う．多角形を描く（円も360角形の近似で描いている）．
CLEAR	変数の初期化，メモリ（テキスト）の上限の設定のみ（IBASICインタプリタ上でのみ）． コンパイラが目的のソース上では，変数の初期化はCLR. $Lで文字列変数の最大値を指定．
CLOSE	なし．
CLS	CLSの引数 なし：テキスト 0　：全グラフィック（B.R.G） 1　：グラフィック1（Blue） 2　：グラフィック2（Red） 3　：グラフィック3（Green） 4　：全グラフィック＋テキスト
〔1〕COLOR （各部の色）	書式が違う．
〔2〕COLOR （カラーパレット）	PALET．パレット，カラー．
COLOR@	なし．
COM ON/OFF/STOP	なし．
CONSOLE	同じ．
CONT	同じ．
COPY	なし．
DATA	同じ．
DEF FN	なし．
DEF INT/SNG/DBL/STR	なし．
DEF USR	なし．
DELETE	同じ．
DIM	2次元まで．かつ最大254まで．
EDIT	同じ．
END	ファイルのクローズはなし．
ERASE	なし．
ERROR	なし．
FIELD	なし．
FOR～TO～ STEP～NEXT	制御変数が終値でなければループから離脱できない（N88はできる）． NEXTには1変数のみが対応（NEXT A，B は不可．NEXT A：NEXT B とする）．
GET	なし．
GET@	なし．
GOSUB～RETURN	参照先ラベルの指定が違う． ex．GOSUB "GOOD"（N88→GOSUB＊GOOD） RETURNに行番号指定は不可．戻る番地を指定する場合，CANCELしてGOTOで飛ばす． ex．CANCEL：GOTO 200（N88→RETURN 200）
GOTO/GO_TO	GO_TOは不可．ラベル（GOSUBと同じ）．
HELP ON/OFF/STOP	なし．

N88-BASIC	dB-IBASIC
IF～THEN～ELSE IF～GOTO～ELSE	THEN以下が文の場合，THEN省略可． ex. IF P<>0 PRINT P IF～ GOSUBも可．
INPUT	ひとつのINPUT文に対し１変数が対応(複数の変数は指定できない)．
INPUT#	なし．
INPUT WAIT	なし．
KEY	同じ．
KEY LIST	同じ．
KEY ON/OFF/STOP	なし．
LET	不可．直接代入する． ex. A=0 (N88→LET A=0)
LINE	書式が違う．ラインスタイル(ラインパターン)の指定方法が違う．STEPは不可．
LINE INPUT	なし．
LINE INPUT#	なし．
LINE INPUT WAIT	なし．
LIST/LLIST	表示スクロールの一時停止はスペースキー． (タイミングによって止まらない場合もある)
LOAD"CAS:"	LOAD ファイルディスクリプタなし.Rオプション不可．
LOAD?"CAS:"	LOAD? ファイルディスクリプタなし．
LOCATE	カーソルスイッチ指定なし．
LSET/RSET	なし．
MID$	関数のみ．ステートメント（文字列の一部の書き替え）機能はなし．
MON	同じ．
MOTOR	なし．
NEW	同じ．
NEW ON	なし．(BOOTしてN88-BASICモード上で)
ON COM GOSUB	なし．
ON ERROR GOSUB	なし．
ON HELP GOSUB	なし．
ON～GOSUB/ ON～GOTO	GOSUB/GOTO参照．
ON KEY GOSUB	なし．
ON PEN GOSUB	なし．
ON STOP GOSUB	なし．
ON TIME$ GOSUB	なし．
OPEN	なし．
OPTION BASE	なし．
OUT	同じ．
〔1〕PAINT	座標，境界色の指定方法が違う．STEPは不可．
〔2〕PAINT	STEPは不可．タイルパターン，境界色の指定方法がが違う．バックグランド不可．
PEN ON/OFF/STOP	なし．
POINT	STEPは不可．
POKE	同じ．
PRESET	STEPは不可．
PRINT	同じ．
LPRINT	なし．
PRINT#	PRINT#0，CRT画面に対してのみ．

N88-BASIC	dB-IBASIC
PRINT USING	なし.
PSET	STEPはなし.
PUT	なし.
PUT@	なし.
RANDAMIZE	なし.
READ	同じ.
REM	同じ.
RENUM	同じ.
RESTORE	ラベルの指定方法が違う.
RESUME	なし.
RUN	同じ.
SAVE	ファイルディスクリプタなし. ファイル名のみ13文字以内で指定. ex. SAVE "ABC" ⏎
SCREEN	書式, 機能が違う. 白黒モードなし. 高速書き込みスイッチなし.
STOP	同じ.
STOP ON/OFF/STOP	なし.
SWAP	なし.
TERM	なし.
TIME$ ON/OFF/STOP	なし.
TRON/TROFF	なし.
VIEW	なし.
WAIT	機能が違う. ポートのモニタなし. 指定時間の実行停止.
WIDTH	CRT画面のみ.(テキスト)
WIDTH LPRINT	なし.
WINDOW	なし.
ABS	同じ.
ASC	同じ.
CSRLIN	POS(1). 引数0のときは桁位置. ex. Y=POS(1) (N88→Y=CSRLIN)
CHR$	同じ.
CINT/CDBL/CSNG	なし.
COS/SIN	書式が違う. 単位は度. 整数型BASICのため,あらかじめ倍数を与えておく. ex. A=COS(60,100)
CVI/CVS/CVD	なし.
DATE$	なし.
EOF	なし.
ERL/ERR	なし.
EXP	なし.
FIX	なし.
FPOS	なし.
FRE	書式が違う. 空テキストエリアのみ.
HEX$	同じ.
INKEY$	INKEY$(2)
INP	同じ.
LEFT$	同じ.
LEN	同じ.
LOC	なし.

N88-BASIC	dB-IBASIC
LOF	なし.
LOG	なし.
LPOS	なし.
MAP	なし.
MID$	同じ.（関数として）
MKI$/MKS$/MKD$	なし.
OCT$	なし.
PEEK	同じ.
PEN	なし.
〔1〕POINT	なし.（Last Referenced Pointの値）
〔2〕POINT	同じ.（グラフィックのドット）
POS	POS(0). 式はダミーではない. {0：水平 {1：垂直
RIGHT$	同じ.
RND	書式，機能が違う. 0～式 -1までの乱数. ex. RND(5) →0, 1, 2, 3, 4
SGN	同じ.
SPACE$	同じ.
SPC	機能が違う. 行の先頭（0桁目）からのスペース.
SQR	値は整数部のみ.
STR$	正の数値の場合，頭のスペースは無視. ex. PRINT LEN(STR$(123)) 3 (N88→4を返す)
STRING$	同じ.
TAB	同じ.
TIME$	TIME.
USR	ステートメント.
VAL	同じ.
VARPTR	なし.
VIEW	なし.
WINDOW	なし.

dB-IBASICメモリマップ



I-BASICが起動中，テキストウインドウは8000(H)～83FF(H)RAMになります．
疑似PCGを使う場合は CLEAR &H77FF を実行します．

アルファベット順索引

A ABS …… 18
APPEND …… 4
ASC …… 19
AUTO …… 4
B BEEP …… 17
BOOT …… 4
BOX …… 12
C CALL …… 16
CANCEL …… 7
CASE END …… 7
CASE FALSE OF …… 7
CASE TRUE OF …… 7
CHR$ …… 20
CIRCLE …… 12
CLEAR …… 7
CLR …… 7
CLS …… 12
CMD …… 6
COLOR …… 13
CONSOLE …… 12
CONT …… 4
COS …… 19
D DATA …… 8
DEFCHR$ …… 8
DELETE …… 4
DIM …… 8
E EDIT …… 4
END …… 8
F FIND …… 5
FOR…TO…STEP～NEXT …… 9
FRE …… 17
G GOSUB …… 9
GOTO …… 9
GPRINT …… 14
GPUT …… 14
H HEXCHR$ …… 19
HEX$ …… 20
I IF～THEN…(ELSE)/IF～GOTO/GOSUB…(ELSE)… 9
INKEY$ …… 20
INPUT …… 11
INSTR …… 20
K KEY …… 16
KEY LIST …… 5
L LEFT$ …… 20
LEN …… 20
LINE …… 13
LIST/LLIST …… 5
LOAD …… 5
LOAD ? …… 5
LOCATE …… 13
M MEM …… 11
MID$ …… 21
MIRROR$ …… 21
M MON …… 5
MPUT …… 14
N NEW …… 5
O ON～GOSUB …… 9
ON～GOTO …… 9
OUT …… 11
P PAINT …… 15
PALET …… 14
PATTERN …… 15
PEEK …… 17
PEEK@ …… 17
PEEW …… 18
POINT …… 18
POKE …… 11
POKE@ …… 11
POKW …… 11
POS …… 18
POSITION …… 14
PRESET …… 16
PRINT …… 12
PRINT#0 …… 12
R READ …… 9
REM …… 10
RENUM …… 6
REPEAT～UNTIL …… 10
RESTORE …… 10
RETURN …… 10
RIGHT$ …… 21
RND …… 19
RUN …… 6
S SAVE …… 6
SCREEN …… 16
SCRN$ …… 21
SGN …… 19
SIN …… 19
SOUND …… 17
SPACE$ …… 21
SPC …… 18
SQR …… 19
STICK …… 22
STOP …… 10
STRIG …… 22
STRING$ …… 21
STR$ …… 21
T TAB …… 18
TIME …… 22
U USR …… 16
V VAL …… 21
W WAIT …… 17
WIDTH …… 16

$L …… 6
＊“ラベル名” …… 10

PC-8801mkII／PC-8801

dB-IBASIC リファレンス・ハンド・ブック

発行日　昭和59年2月1日　第1版　第1刷

企画制作
著　　作　デービーソフト技術部

発行所　デービーソフト株式会社

〒060 札幌市中央区北3条西2丁目
カミヤマビル☎(011)251-7462

編　集／デービーソフト企画室

御注意

①本書の内容の一部又は全部を無断で転載することは固くお断わりします。
②本書の内容に関して、将来予告なしで変更することがあります。
③本書の内容については万全を期して作成いたしましたが、万一御不審な点や誤り、記載もれなど、お気付きの点がありましたら御一報下さい。
④本プログラムを運用した結果の影響につきましては、責任を負いかねますので御了承願います。

dB-SOFT

企画制作
著　作　デービーソフト
〒060 札幌市中央区北3条西2丁目カミヤマビル4F ☎(011)251-7462

N3-L0022

COPYRIGHT © 1984 dB-SOFT INC. NO PART OF THIS PROGRAM SHOULD BE USED WITHOUT PERMISSION.